

(12) UK Patent Application (19) GB (11) 2 387 453 (13) A

(43) Date of A Publication 15.10.2003

(21) Application No 0200277.2

(22) Date of Filing 08.01.2002

(71) Applicant(s)
Staffordshire University
(Incorporated in the United Kingdom)
Beaconside, STAFFORD, Staffordshire,
ST18 0AD, United Kingdom

(72) Inventor(s)

J Rees

T Cockle

N Willis

(74) Agent and/or Address for Service

Swindell & Pearson

48 Friar Gate, DERBY, DE1 1GY,
United Kingdom

(51) INT CL⁷
G06F 17/30

(52) UK CL (Edition V)
G4A AMC

(56) Documents Cited
GB 2350213 A WO 2000/023916 A1
US 6094662 A US 5878223 A

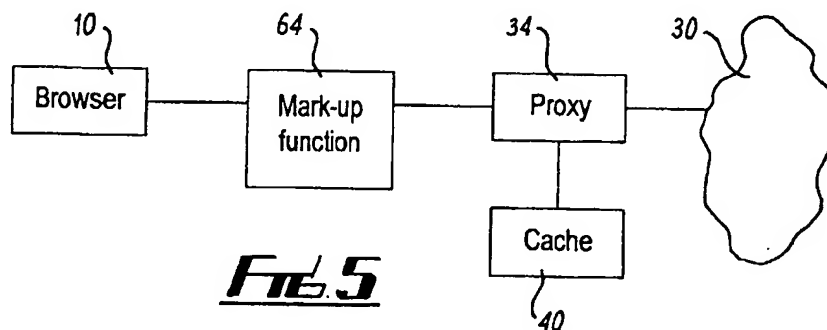
(58) Field of Search
INT CL⁷ G06F
Other: Online: EPODOC, WPI, JAPIO, TDB, XPESP,
INSPEC, IEL

(54) Abstract Title

Cache arrangement indicates when object is available from cache

(57) A user of a client machine 10 uses browser software to gain access to objects on a network 30, through a proxy server 34. The proxy server 34 has cache memory 40 to allow objects to be retrieved from cache memory, when available. A mark-up function 64 is provided to modify links within objects passed to the client 10, so that a user can discriminate between links to objects which are available from cache memory 40, and links to objects which are available only from the network 30. This improves the efficiency with which cache memory is used.

The objects are preferably HTML documents. The cached links may be indicated to the user by an associated graphic.

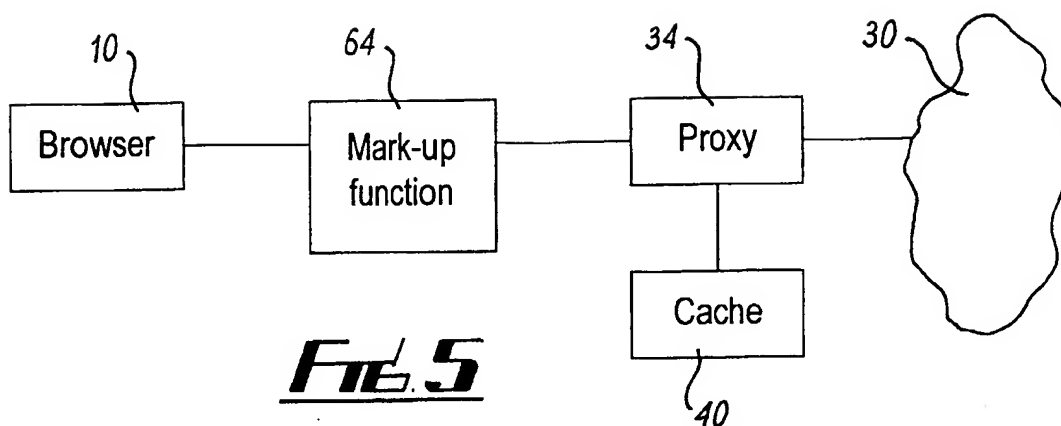


At least one drawing originally filed was informal and the print reproduced here is taken from a later filed formal copy.

The claims were filed later than the filing date but within the period prescribed by Rule 25(1) of the Patents Rules 1995.

GB 2 387 453 A

3/6

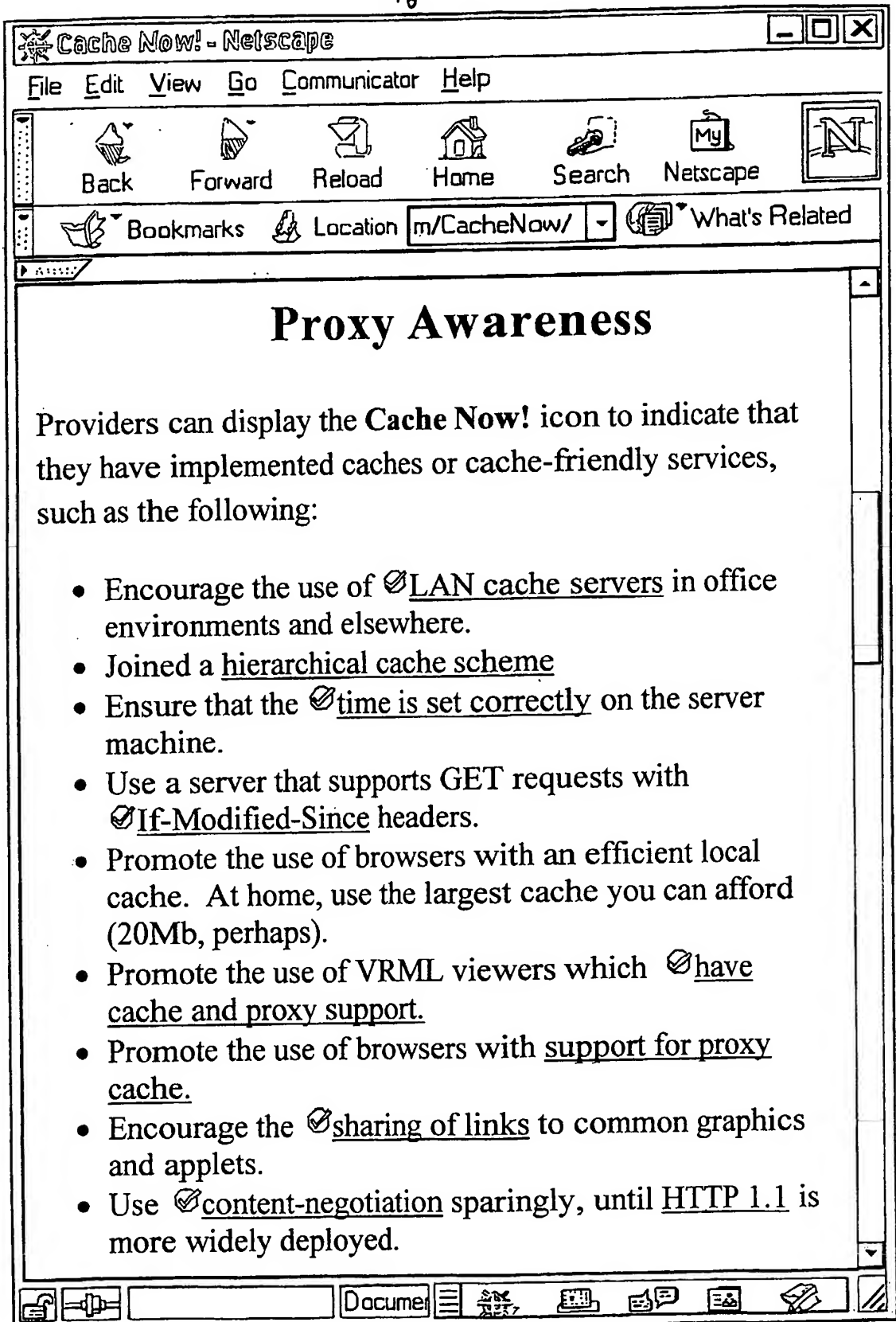


< P > Consider filing a < A HREF = " http : // www.patents.co.uk " > patent application < / A > for your invention < / P >

FIG. 7a

< P > Consider filing a < IMG src = X > < A HREF = " http : // www.patents.co.uk " > patent application < / A > for your invention < / P >

FIG. 7b



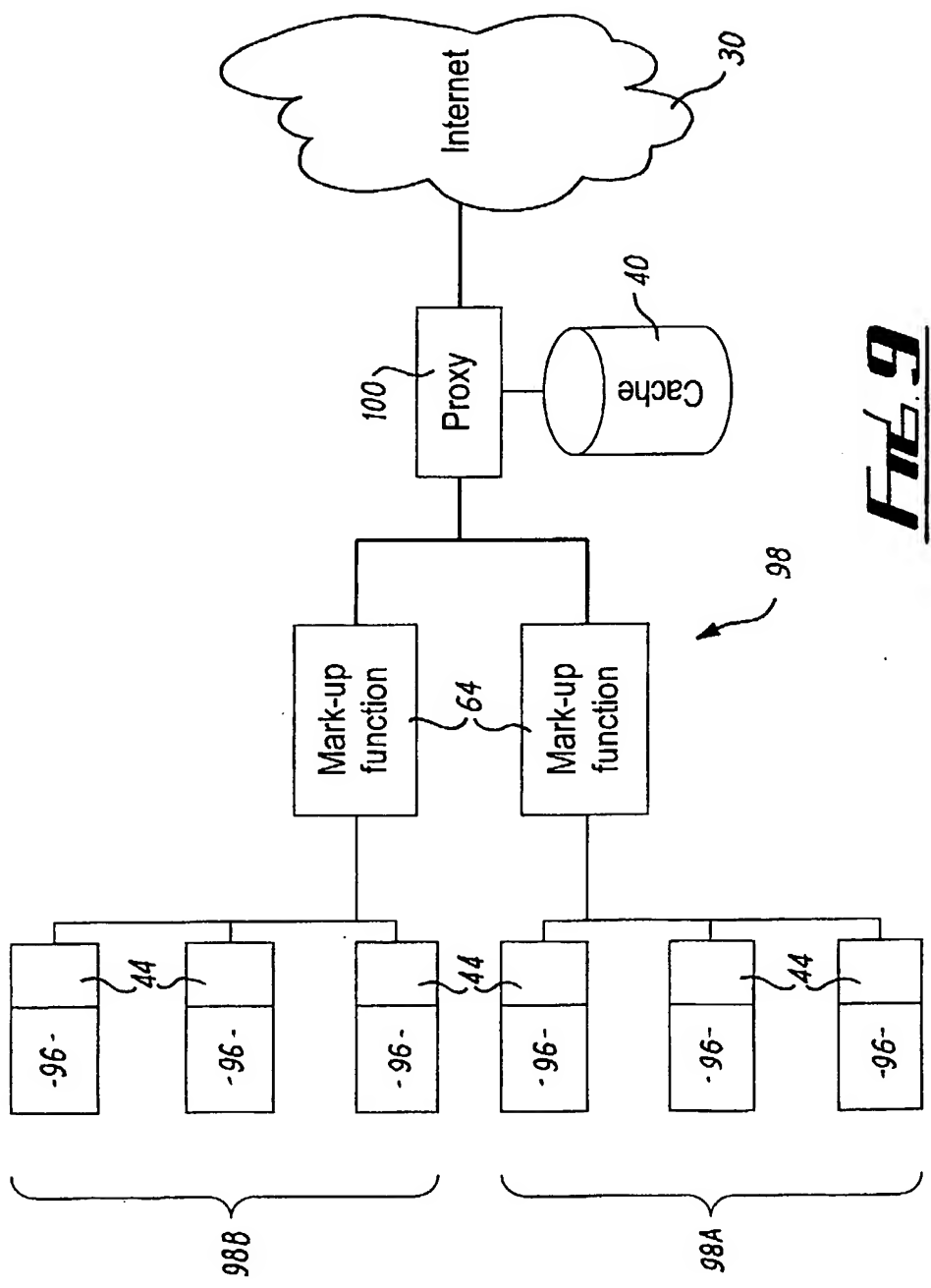


Fig. 9

Cache Arrangement

The present invention relates to improvements in or relating to cache arrangements for use during file retrieval from a computer network.

In one conventional form of computer network arrangement, one or more server machines are connected to the network and used to store files which may be required by users. These servers may be termed "origin" servers, denoting that they are the origin of files to be retrieved over the network. Users gain access to the network by using "client" machines which request files to be delivered to them, from the appropriate origin server, over the network. For example, files stored on origin servers (such as those which form the internet) are delivered to a client machine in order to be viewed by a user, in fulfilment of requests made by that user by means of the client machine. In this example, the client machine is usually not connected directly to the network (the internet), but is connected to a "proxy" server which is itself connected to the internet. Thus, requests are initially passed from the client machine to the proxy server and then directed from the proxy server to the appropriate origin server. Files returned from the origin server are passed on to the client machine, by the proxy server. Files may contain data representing text, images or other information, and are generally called "objects". Proxy servers and/or client machines may be provided with cache memory in which objects retrieved from the network in fulfilment of previous requests are stored in order to fulfil more quickly any future request for the same object. Consequently, when the object is requested again in future, the request can be fulfilled from the contents of the cache memory, without recourse to the origin servers. However, in the event that a requested object is not available from cache memory, the full process of retrieving the object from an origin server must be completed, resulting in increased network traffic and slower request fulfilment than when cache memory is used.

The present invention provides a cache arrangement for a computer network of the type in which object requests are generated by a computer

operating as a client and are passed to a computer operating as a proxy server, and the proxy server locates the object on the network or within cache memory available to the proxy server, and passes the object to the client in fulfilment of the request, the cache arrangement including mark-up means operable to analyse target objects being passed to the client, to identify any links to further objects and to modify target objects to cause the links to appear in the target object in a manner which enables a user to discriminate between links to objects which are available at least in part from cache, and links to objects which are available only from the network.

The mark-up means preferably identifies links by surveying the target object for data or data structures characteristic of a link. The mark-up means is preferably operable to interrogate the cache means when the mark-up means identifies a link, the cache means being operable to respond to interrogation by indicating the presence or absence of the or part of the linked object within the cache. The mark-up means may be operable to interrogate the cache means on each occasion that a link is identified, or in relation to groups of links identified in a target object.

The mark-up means is preferably operable to modify the target object to cause links to cached objects to be displayed in a manner visibly distinct from the manner of display of links to uncached objects. Preferably the mark-up means modifies links available from cache. Preferably the mark-up means is operable to insert a symbol into the target object to appear in association with links to cached objects, when the target file is displayed.

The client may be provided by a machine running client software, the mark-up means being executed, in use, by the client machine. Alternatively, the server may be provided by a machine running server software, and the mark-up means is executed in use by the server machine. In a further alternative, the client may be provided by a machine running client software, the server machine being provided by a machine running server software, and the mark-up means is executed, in use, by a further machine operatively connected to the client machine and to the server machine.

The proxy server may be connected to a public network. The client and server may be connected to a private network.

The cache memory may be associated with the server machine and/or the client machine.

In a second aspect, the invention provides a computer network comprising at least one origin server operable to provide objects in response to object requests, at least one client operable by a user to request objects, and at least one cache means operable to store objects for use in fulfilling future object requests without recourse to an origin server, wherein the cache means is a cache arrangement as defined in any of the preceding six paragraphs.

Preferably at least one proxy server is in communication with the or each origin server, the or each client being connected to the or a proxy server, and to the or each server through the proxy server.

In a third aspect, the invention provides computer apparatus connected, in use, to a computer network and operable, in relation to object requests made by a client connected to the computer network, as mark-up means as defined in any of the definitions relating to the first aspect of the invention.

In a fourth aspect, the invention provides computer software which, when installed on a computer system, is operable as mark-up means as defined in any of the definitions relating to the first aspect of the invention.

The invention further provides a carrier medium carrying computer software as defined in the preceding paragraph.

The carrier medium may be a recording medium. Alternatively, the carrier medium may be a transmission medium, the software being carried by a signal propagating on the transmission medium.

In a fifth aspect, the invention provides an object for display by a

computer connected to a network which includes cache memory, the object containing links to other objects, and wherein the object is modified for display, to cause links to appear in the said object, when displayed, in a manner which enables a user to discriminate between links to objects available at least in part from cache memory, and links to objects available only from the network.

The invention further provides a signal propagating over a transmission medium and representing an object as defined in the preceding paragraph.

Examples of the present invention will now be described in more detail, by way of example only, and with reference to the accompanying drawings, in which:-

Fig. 1 is a schematic diagram of a general purpose computer with which the present invention may be implemented;

Figs. 2 and 3 illustrate the connection of a computer to a network such as the internet, by means of a previous proposal;

Fig. 4 illustrates a screen image created from objects retrieved from the internet;

Fig. 5 illustrates the connection of a computer to a network such as the internet, in accordance with one embodiment of the present invention;

Fig. 6 is a flow diagram for operations in accordance with the present invention;

Fig. 7 is a schematic diagram of data before and after assessment by the flow diagram of Fig. 6;

Fig. 8 is a screen image created in accordance with the present invention; and

Fig. 9 is a schematic diagram of the invention implemented in a commercial environment.

Background

It is convenient first to describe briefly the basic computer hardware by means of which the present invention may be implemented.

Fig. 1 illustrates, in simplified schematic form, a computer 10, which may be a computer of the type known as an IBM compatible personal computer (PC), but the invention can be implemented with other types of computer.

The computer 10 has a processor 12 which runs under software control and has associated main (RAM) memory 14 and auxiliary memory 16, such as a hard disc drive. Various input and output apparatus is connected to the processor. A display screen 18, a keyboard 20 and a mouse 21 (or other cursor control device) provides for entry of data and instructions by a user, and for a user to see the results on the display 18. A disc drive 22 allows data and software to be read from or to a removable memory device or recording medium such as a disc 24. A modem 26 provides means by which the computer 10 may communicate, at 28, with other machines, such as by means of a computer network.

Network Connection and Object Retrieval

Fig. 2 illustrates, schematically, the manner in which a computer 10 of the type described above may be connected to a public network, such as the internet, in accordance with a previous proposal which does not form part of the present invention.

In Fig. 2, a public network, for example the internet, is represented at 30 as a group of origin servers 32. In practice, many thousands of origin servers exist and are connected together by many complex interconnections, as will be known to the skilled reader. Consequently, connections between the servers 32 are not shown.

The purpose of the origin servers 32 is to deliver "objects" to the computer 10 when connected to the internet 30. The computer 10 serves as a client by means of which a user may make requests for objects stored in the origin servers 32. These objects are data files of various types, which will be discussed in more detail below. The client machine 10 is commonly not connected directly to the internet 30, but by means of an intermediary machine

34, called a proxy server. Thus, the client machine 10 has a connection 36 to the proxy server 34. The proxy server 34 has a connection 38 to the internet. Consequently, requests made by the client machine 10 can be passed to the internet 30 through the proxy server 34. Objects retrieved from the origin servers 32 can be passed back through the proxy server 34, to the client machine 10.

In one common arrangement, particularly for domestic users, the proxy server 34 will be owned by a business known as an internet service provider. Many individual users will subscribe to the services of the internet service provider, allowing them to make access by means of a client machine 10 to the proxy server 34 of the internet service provider. The internet service provider maintains the proxy server 34 and the link 38 to the internet 30. In consequence, the connection 38 will carry all of the traffic between the internet 30 and any client machine connected to the proxy server 34. In the case of an internet service provider with many subscribers, the connection 38 will therefore be required to have much greater bandwidth than the bandwidth required for any connection 36 between an individual client machine 10 and the proxy server 34.

Unfortunately, even with very large bandwidths for the connection 38, it is found in practice that heavy traffic, such as at certain times of the day or night, can significantly reduce the time taken to retrieve objects from the internet servers 32. This can lead to frustration on the part of users.

In consequence, a practice has grown up of providing cache memory 40, associated with the proxy server 34. The cache 40 is used to store objects retrieved from the servers 32 in fulfilment of a request. Future requests for the same object (by the same or a different subscriber) can therefore be fulfilled from the contents of the cache 40, rather than from the origin servers 32, thus avoiding the need to communicate over the link 38 and speeding up the process of fulfilling a request for an object.

In practice, the capacity of any cache 40 will be finite and thus, it is

necessary for the proxy server 34 to operate the cache 40 in accordance with an algorithm which determines what is cached, and how and when to select objects for discard from the cache 40. One simple algorithm detects when the cache 40 is full, and then discards the oldest object within the cache, to make space for further objects.

It has been found that the effectiveness of caches, in practice, may be significantly inferior to the effectiveness expected from theoretical analysis. We have realised that one factor which hinders efficient use of cache is that a user is not aware of the contents of the cache 40 and is unaware which requests are fulfilled by recourse to the cache 40, and which are fulfilled by recourse to the origin servers 32. The present invention makes use of this realisation, as will be described below.

Browser Software and Objects

Figs. 3 and 4 allow the operation of a client machine 10 to be described in more detail, and the nature of objects retrieved from internet servers 32 to be described in more detail.

In Fig. 3, the client machine 10 is illustrated in simplified form to show only the elements relevant to the functions to be described.

In Fig. 3, an area 42 represents the processing operations of the machine 10, provided by the processor 12 and software residing in the memory space associated with the processor 12 (primarily the RAM 14). This provides functions represented as a browser 44 and a control function 46. The control function 46 provides all interface requirements between the browser 44 and the various input/output devices described above, and between the browser 44 and other functions within the machine 10, and provides all other control required within the machine 10. In particular, the control function 46 interfaces between the browser 44 and the keyboard 20, display 18 and modem 26.

The modem 26 provides a connection from the machine 10, for example

to a proxy server 34 in the arrangement shown in Fig. 2. The keyboard 20 and mouse 21 allow a user to enter instructions, such as the identity of an object required to be retrieved for display. Objects retrieved may be displayed on the display 18. In practice, operation of the machine 10 by a user will be interactive, in that instructions given to the machine will be affected by what is shown on the display 18, which may be an object previously retrieved.

Fig. 4 illustrates, schematically, an example of an image created on the display 18 after objects have been retrieved in fulfilment of a user request. In this example, the image is a web page.

A user may request the retrieval of an object either by entering an address for the object in a standard form, known in relation to the internet as a Uniform Resource Locator (URL) or by using a hyperlink (see below). Either operation causes the client machine 10 to send a request to the proxy server 34, to be met by recourse to the internet 30 and the origin servers 32, or from cache memory 40. In prior proposals, the user has been unaware which alternative has been used to fulfil the request and the cache function can thus be considered transparent, in that the existence and operation of the cache is not apparent to the user.

Web pages are written in Hyper Text Mark-Up Language (HTML). This is a language which adds many enhancements to plain text. Most importantly, HTML allows the writer to insert hyperlinks into the text. These provide a method for linking one page to another, so that a user can easily move from one web page to a related web page by using the link. Web pages are viewed using software called a web browser. Web pages are stored on origin servers, which are accessed through the internet using the Hyper Text Transport Protocol (HTTP). The location of a web page (i.e. the identity of the origin server holding it, and the appropriate address within the origin server) can be expressed as a Uniform Resource Locator (URL) or as a direct web address. To view a web page a user can enter a URL into the browser. The browser then initiates a series of HTTP transactions. For each transaction the browser fetches a web object from the appropriate origin server (or from a cache). A web object is a file obtainable

from the web (for example a HTML file, a picture or sound clip). One object, such as an HTML file, may identify other objects, such as an image, to be "embedded" within it, to form part of the web page seen by the user. When viewing a web page the browser first downloads the HTML file, then for each embedded object (e.g. a picture) downloads the required file.

In Fig. 4, the outer rectangular boundary 48 represents the main HTML object, which in turn defines two frames 50,62 that form a web page. The user will perceive the boundary of 48 to be the boundary of the web page. Into each frame 50,62 defined by the object 48, a further HTML file is to be loaded. The HTML code within the object 48, defining the objects 50,62, also identifies the internet location of the HTML objects 50,62. These can therefore be retrieved for incorporation into the web page. Into the right hand frame 50, the HTML object "content" is loaded. Into the left hand frame 62, the HTML object "favourites" is loaded.

The HTML object 50 "content" in turn defines the location of two other embedded objects. Firstly, a picture 52 is loaded. The three HTML objects so far described (48, 50 and 62) and the picture 52 all include HTTP headers to describe themselves as cacheable and can be retained in cache memory. The second object to be embedded into the HTML object 50 ("content") is a graphical object 56, for example current share and stock prices. This object is dynamically created each time the web page is displayed. Due to the dynamic nature of the information it conveys, the object 56 identifies itself as being non-cacheable.

The final object to be embedded in this example is the HTML object 62 "favourites". This HTML object (62) does not define any further embedded objects but does describe a set of hyperlinks to further web pages, some of which may already be held in the cache memory.

Improved Cache Arrangement

Fig. 5 illustrates an arrangement in accordance with the present

invention and which differs from the simple arrangement of Fig. 2 primarily by the provision of an additional function at 64. This function is referred to herein as the "mark-up" function, for reasons which will become apparent. The mark-up function 64 is interposed between the browser 44 and the proxy server 34, so that traffic between the browser 44 and proxy server 34 passes through the mark-up function 64. This functional arrangement can be provided in many ways. For example, the mark-up function 64 could be provided by additional software within the client machine 10, or by additional software within the proxy server 34, or by means of a further computer (such as an IBM compatible personal computer or similar) connected to the client machine 10 and the proxy server 34 and executing only the mark-up function 64. In view of the readily understood alternative physical arrangements, the arrangements of Fig. 5 will be described only in relation to the functional connections. Other examples of implementation will be described later in this description.

The principal purpose of the mark-up function 64 is to ensure that the user of the machine 10 is provided with information allowing the user to discriminate between hyperlinks to objects available, at least in part, from the cache memory 40, and hyperlinks to objects available only from the origin servers 32. The cache operations therefore become "opaque", in that the user is presented with information about the existence and contents of the cache, and the source from which a request will be fulfilled.

Sequence of Operation

Fig. 6 illustrates a sequence of steps by which this may be achieved. At 66, the browser 44 detects a user request for an object, either by direct input of a URL, or by operation of a hyperlink. The mark-up function 64 forwards the request at 68 to the proxy server 34. At 70, the proxy server 34 fulfils the request by retrieving the requested target object, either by recourse to the internet 30 and origin servers 32, or by recourse to the cache memory 40. The target object is forwarded to the mark-up function 64 from the proxy server 34.

The mark-up function 64 determines at 72 if the target object is an HTML

object or a non-HTML object such as an image. If the object is a non-HTML object, it is forwarded at 74 to the browser 44 for display to the user. This concludes the HTTP transaction for that object, at 76.

If an HTML object is identified at step 72, the mark-up function 64 begins to interrogate the HTML file at 78, looking for an HTML symbol which identifies a hyperlink. Standard HTML symbols readily allow hyperlinks to be identified in this way.

The mark-up function 64 continues to scan through the target object until all hyperlinks and their associated URL's have been identified. This generates a list of the associated URL's.

The mark-up function 64 then considers each URL in the list, beginning at step 80. Initially, the first URL in the list is used as the basis of a query sent to the cache memory 40 in an appropriate form, such as Intercache Protocol (ICP) language, to interrogate the cache memory 40 as to the presence or absence within the cache memory 40 of the object corresponding to the URL.

The response from the cache memory 40 is considered by the mark-up function 64 at 82. If the cache memory 40 indicates that the object is present, i.e. that a HIT has been scored, the mark-up function 64 acts to modify the target object from which the URL list was constructed, to insert an additional symbol by the URL or hyperlink of each corresponding object which is available from cache. The purpose of these additional symbols will become apparent below, but can be simply summarised as marking up the target object to identify objects available from cache.

The mark-up function 64 goes on to determine at 86 if the URL list has been exhausted and if not, returns to step 80 to repeat the sequence in relation to the next URL in the list.

Once the cache memory 40 has been interrogated in relation to all URL's on the list, the mark-up function 64 moves on to step 88, in which the HTML

object is sent to the machine 10, for the browser 44. The target object is now in modified form, the HTML containing an additional symbol associated with every URL which is within the HTML and which refers to an object available from the cache memory.

This completes the HTTP transaction at 76.

Results

Fig. 7a illustrates schematically the effect of the operation which has just been described. Fig. 7a shows a simple example of HTML code from a target object, in the form in which it is received by the mark-up function 64, i.e. prior to analysis and modification. In this example, the phrase "Consider filing a patent application for your invention" incorporates a hyperlink based on the phrase "patent application", which will be underlined in the web page to specify a hyperlink to the web page at www.patents.co.uk. The presence of this hyperlink is indicated by the tags <A and /A>.

Fig. 7b represents the same HTML code after the process of Fig. 6 has been completed, and assuming that the opaque cache 64 discovers that the object corresponding to hyperlink address www.patents.co.uk is present within the cache memory 40. In consequence, the code relating to the hyperlink is preceded by the additional code because the object at www.patents.co.uk is available from cache memory 40. The additional symbol is used in creating the final web page display for the user, as follows.

Final Display

The additional symbol included in the HTML code 92 is used by the browser 44 when displaying the object which has been retrieved.

Fig. 8 illustrates an example of the result obtained by displaying an HTML object retrieved by means of the opaque cache 64. The screen illustrated in Fig. 8 is generated by the browser software package called Netscape Navigator

(Version 4.7), available from Netscape Inc.

Within the image 94 of Fig. 8, various text is underlined to represent a hyperlink to another object. This might be an object on another website, another object or website dealing with that subject or the like. The reader can readily see that some of the hyperlinks identified in the image 94 are preceded by a large round symbol in addition to being underlined, while others are underlined but not preceded by the round symbol. The round symbol is present only when the corresponding hyperlink has been flagged by the opaque cache as being available from cache memory 40.

The visual symbol is present because the additional HTML symbol () introduced by the mark-up function 64 is an HTML instruction to display an image identified (as an image file cache.gif) in the HTML symbol, and which therefore appears alongside the corresponding hyperlink which the mark-up function 64 has identified as being available from cache memory 40. Every hyperlink to a cached object is therefore marked-up as such, when the page is viewed by the user.

Alternatively, an additional symbol could be displayed to identify a hyperlink which is not available from cache, thus warning the user of the expected slower retrieval.

Consequently, the user is readily able to identify those links which will provide fast retrieval from cache, and those which will require recourse to the internet and can thus be expected to be slower. The user is not prevented from a free choice of any of the links presented, but is able to make an informed choice based on expected retrieval times.

Commercial Network Arrangements

The examples given above have been described primarily in terms of a single client machine with access to a proxy server connected in turn to the internet. This most commonly represents an

individual domestic user who subscribes to an internet service provider service.

In many commercial applications, such as within a commercial organisation, many users will be connected to a private network within the organisation, and internet access will be achieved by means of a server on the private network being externally connected for internet access. The invention described above can readily be implemented in this context, for instance in the manner illustrated in simple form in Fig. 9.

In the arrangement of Fig. 9, a number of user computers 96, each provided with a browser 44 are connected to a private network 98, forming two groups 98A,98B which might represent different departments or locations, for example. A server 100 is also connected to the network 98 and provides a private server for the network 98, and also acts as an internet proxy server. Consequently, the server 100 has associated cache memory 40. Connections from the groups 98A,98B to the proxy server 100 are through a mark-up function 64 in the manner described above.

Requests from users are thus first propagated over the network 98 to the server 100, through the mark-up function 64, and to the internet 30 as required. Replies are delivered through the mark-up function 64 from the server 100 and thus over the network 98 to the user computer 96.

The mark-up function 64 and cache memory may be provided in various arrangements or combinations. The server 100 may be provided with cache memory 40, as described above in relation to Fig. 5. Alternatively, or in addition, cache memory may be provided in association with the mark-up function 64, or elsewhere on the network 98. In particular, if the mark-up function 64 is provided within the server 100, or within the premises owned by the owners of the network 98, it is particularly preferred to provide cache memory within the operator premises. This allows requests from within the organisation to be fulfilled, wherever possible, from cache memory within the organisation, thus reducing the traffic out to the internet 30 and thus reducing the bandwidth required for this traffic, and thus the cost of internet

connections from the organisation.

Advantages of the Arrangements Described Above

The various arrangements described above are expected to provide practical and economic advantages or combinations of advantages, as follows.

Users are provided with information allowing them to choose between links with fast access from cache, and links requiring downloading from the internet and thus likely to be slower. This can reduce the frustration of impatient users. In addition, the fact that an object is available from cache will indicate that it has previously been requested. In the commercial environment illustrated in Fig. 9, this information may be valuable to a user as indicating that a user within that organisation has previously made use of that object, thus suggesting that the object is adequate for their purpose. In other situations, the presence of an object in cache can be used as an indication of popularity, leading to a reinforcement of that popularity by encouraging additional requests.

It is thus to be expected that users will make more use of cached resources than they would be able to do without this additional information. This brings advantages to the owners of the networks, particularly commercial private networks of the type illustrated in Fig. 9. External internet traffic can be reduced, thus reducing the cost of bandwidth required externally of the organisation. Efficiency and consistency within the organisation may be improved by encouraging users to access objects previously used by others within the organisation. More refined use of cache resources, by encouraging users to use cached resources rather than fresh resources, is expected to provide very significant reductions in the amount of disc space required for cache storage, thus reducing the capital and running costs of systems.

The arrangement also provides a simple means for content control. For example, having identified resources available from cache by means of the opaque cache arrangement described, an organisation can readily implement

systems to prevent access to resources which have not been cached. For example a school could readily ensure that pupils have access only to objects previously vetted and cached, but not to other objects from the internet.

Modifications

Modifications can be made to the apparatus and arrangements described above, without departing from the scope of the present invention. In particular, many hardware technologies can be used to implement the various components of the systems described. Software required to operate the systems can be created in many different ways and languages, as will be apparent to the skilled reader.

The mark-up function for modifying a conventional cache arrangement of the type illustrated in Fig. 2 can readily be provided by separate hardware and associated software, for connection between a user and proxy server as illustrated in Fig. 5. In this alternative, the additional hardware may incorporate memory for use as cache memory, or the function may interrogate cache memory held elsewhere (at the proxy server or client), or both. This arrangement is relatively easy to implement and does not unduly increase the workload of the proxy server, but requires an additional machine and further network traffic between the mark-up function and the cache memory if located elsewhere, leading to a reduction in file retrieval speeds (or increased latency).

Alternatively, the mark-up functions could be provided for existing arrangements by installing appropriate additional computer software (at the client or proxy machine) operable to mark-up HTML files in the manner described above. For example, additional software could provide the mark-up function locally at the proxy server, or locally at the client. In the former case, the mark-up function imposes no additional workload on the proxy server, other than to respond to enquiries if cache memory is at the proxy server, but requires this additional software to be installed on all client machines, and results in an increase in network traffic in the event that the cache memory is stored elsewhere than on the client machine. This increased network traffic is

reduced if the additional mark-up function software is installed on the server, if the server contains the cache memory, and this arrangement also allows a single installation to be available to all clients of that proxy server. However, the workload of the proxy server is consequently increased.

This additional software could be provided on a carrier medium such as a recording medium (a disc such as a magnetic, optical or other data storage device) or by transmission as a signal propagating on a transmission medium. Thus, the software could be downloaded for installation.

The mark-up function could be provided as part of the browser software running on the client machine, or the server software running on the proxy server, but this would require all client machines to have updated browsers installed, or the proxy server software to be updated.

It should be noted that in all of the examples described above, there has been reference to a single cache memory but it is to be understood that this is symbolic for the entire cache memory space available to a particular user, however that might be broken up into individual memory devices, and whatever the location of those individual memory devices might be. Thus, when the mark-up function is interrogating cache memory, it may be necessary to send more than one enquiry, so that each part of the available cache space is interrogated.

An object which has been modified in accordance with the procedures described above will differ from an object which has not benefited from the present invention, and thus itself represents an embodiment of the invention, as does a signal propagating over a transmission medium and representing an object which has been modified in this way.

Many references have been made to HTML language. A mark-up function could be implemented in relation to other languages in an analogous fashion. For example, techniques according to the invention could readily be implemented in XML language.

Various machines have been described as clients, servers etc. These are functional descriptions representing the relevant function when implementing the present invention. The multi-function nature of modern computer systems, and modern network techniques mean that a machine being used as a client or server in accordance with the present invention might also be used (simultaneously or at other times) as a server or client for other purposes not connected with the present invention.

Whilst endeavouring in the foregoing specification to draw attention to those features of the invention believed to be of particular importance it should be understood that the Applicant claims protection in respect of any patentable feature or combination of features hereinbefore referred to and/or shown in the drawings whether or not particular emphasis has been placed thereon.

CLAIMS

1. A cache arrangement for a computer network of the type in which
5 object requests are generated by a computer operating as a client and are
passed to a computer operating as a proxy server, the proxy server locates
the object on the network or within the cache memory available to the proxy
server, and passes the object to the client in fulfilment of the request, the
cache arrangement including mark-up means operable to analyse target
10 objects being passed to the client, to identify any links to further objects and to
modify target objects to cause the links to appear in the target object in a
manner which enables a user to discriminate between links to objects which
are available at least in part from cache memory and a link to objects which
are available only from the network.
- 15
2. An arrangement as claimed in claim 1 wherein the mark-up means
identifies links by surveying the target object for data or data structures
characteristic of a link.
- 20
3. An arrangement as claimed in claim 1 wherein the mark-up means is
operable to interrogate the cache memory when the mark-up means identifies
a link, the cache arrangement being operable to respond to interrogation by
indicating the presence or absence of the or part of the linked object within the
cache memory.
- 25
4. An arrangement as claimed in any of claims 1, 2 or 3 wherein the
mark-up means is operable to interrogate the cache memory on each
occasion that a link is identified or in relation to groups of links identified in a
target object.
- 30
5. An arrangement as claimed in any preceding claim wherein the mark-
up means is operable to modify the target object to cause links to cache
objects to be displayed in a manner visibly distinct from the manner of display
of links to uncached objects.

6. An arrangement as claimed in any preceding claim wherein the mark-up means modifies links available from the cache arrangement.
- 5 7. An arrangement as claimed in any preceding claim wherein the mark-up means is operable to insert a symbol into the target object to appear in association with a link to cached objects when the target file is displayed.
8. An arrangement as claimed in any preceding claim wherein the client is
10 provided by a machine running client software, the mark-up means being executed, in use, by the machine running client software.
9. An arrangement as claimed in any of claims 1 to 8 wherein the proxy server is provided by a machine running server software and the mark-up
15 means is executed, in use, by the machine running server software.
10. An arrangement as claimed in any of claims 1 to 9 wherein the client is provided by a machine running client software, the server machine being provided by a machine running server software and the mark-up means is
20 executed, in use by a further machine operatively connected to the client machine and to the server machine.
11. An arrangement as claimed in any preceding claim wherein the proxy server is connected to a public communication network.
25
12. An arrangement as claimed in any preceding claim wherein the client and the server are connected to a private communications network.
13. An arrangement as claimed in any preceding claim wherein the client
30 and proxy server are provided by a single appropriately configured machine.
14. An arrangement as claimed in any preceding claim wherein the cache memory is associated with the proxy server and/or the client.

15. An arrangement as claimed in any preceding claim wherein the cache memory is provided by a plurality of machines associated in an appropriate memory architecture to provide a consolidated memory formed from respective memory elements of each machine constituting the cache memory,
5 each machine connected to other machines in a local area network (LAN) to allow interrogation of all memory elements of the cache memory located upon each respective machine.
16. A cache arrangement substantially as hereinbefore described with
10 reference to the accompanying drawings.
17. A computer network comprising at least one origin server operable to provide objects in response to object requests, at least one client operable by a user to request objects and at least one cache means operable to store
15 objects for use in fulfilling future object requests without recourse to an origin server, wherein the cache means is a cache arrangement as claimed in any preceding claim.
18. A computer network as claimed in claim 17 wherein at least one proxy
20 server is in communication with the or each origin server, the or each client being connected to the or a proxy server, and to the or each origin server through the proxy server.
19. A computer network substantially as hereinbefore described with
25 reference to the accompanying drawings.
20. Computer apparatus connected, in use to a computer network and operable, in relation to object requests made by a client connected to the computer network, as mark-up means as defined in any of claims 1 to 18.
30
21. Computer apparatus connect, in use, to a computer network and operable substantially as hereinbefore described with reference to the accompanying drawings.

22. Computer software which, when installed on a computer system, is operable as a mark-up means as defined in any of claims 1 to 18.

23. Computer software which, when installed on a computer system is operable substantially as hereinbefore described with reference to the accompanying drawings.

24. A carrier medium carrying computer software as defined in claim 24.

25. A medium as claimed in claim 25 wherein the carrier medium is a recording medium.

26. A medium as claimed in claim 25 wherein the carrier medium in a transmission medium, the software being carried by a signal propagated on the transmission medium.

27. An object for display by a computer connected to a network which includes cache means, the object containing links to other objects, and wherein the object is modified for display, to cause links to appear in the said object, when displayed, in a manner which enables a user to discriminate between links to objects available at least in part from cache memory and links to objects available only from the network.

28. An object for display by a computer connected to a network substantially as hereinbefore described with reference to the accompanying drawing.

29. A signal propagating over a transmission medium and representing an object as defined in claims 27 or 28.

30. Any novel subject matter or combination including novel subject matter disclosed herein, whether or not within the scope of or relating to the same invention as any of the preceding claims.